



Description: This AppNote will explain you the Data Mining Quick Start.

OS Requirement: Windows 2000,2003,XP

General Requirement: Genesis V 8.0 installed.

1. Data Mining OPC Server

The Database mining OPC server can be configured using two configuration tools.

1. First is full featured standalone application **Data Mining Configurator** accessible from “**ICONICS GENESIS-32\Tools**” program group. Snapshot of the configurator’s layout is shown on the picture below.

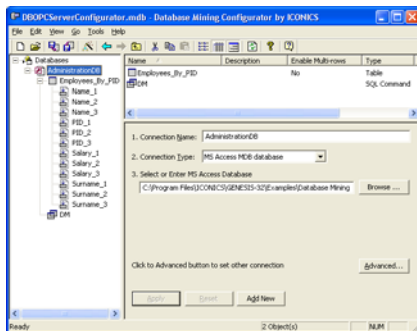


Figure 1

2. Second tool is embedded in **Unified Data Browser on page Database Access** and is available together with OPC DA browser. It was designed for ease of use to allow users to make on-the-fly modifications in the server address space. This second tool supports a subset of features available in Data Mining Configurator tool. The following picture shows the Database Access page in Unified Data Browser.

3.

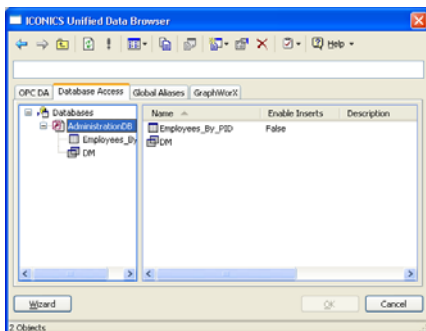


Figure 2

Note: Both tools are capable of configuring Data Mining OPC Servers on local machine only.

1.1 Configuring the Server

First of all the user should decide what type of action she/he wants to do with databases – there are two possible options:

- Read the data periodically and display them in an OPC client (e.g. in GraphWorX)
- Manipulate database data on user request.

The first option is represented by *Data Sources*; the second option by *Data Manipulators*. Both of these objects need to specify a *Database Connection*. For instruction on how to create each of these objects in configuration database see chapters below.

1.2 Database Connections

The Database Connection object is equivalent to a physical database connection. To correctly configure database connection means to set up database type and fill in appropriate number of connection parameters. The following is the list of supported database connections:

1. MS Access
2. MS Excel (limited support given by MS Jet engine)
3. MS SQL (MSDE) server
4. Oracle SQL server
5. ODBC drivers

The following picture shows connection dialog layout for the MS SQL server.

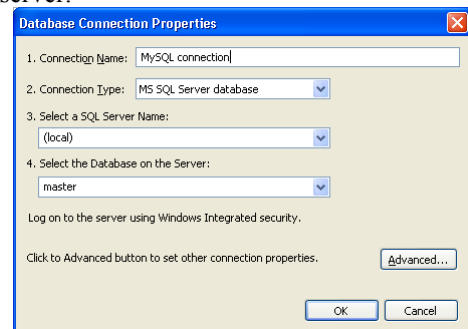


Figure 3

The “**Advanced**” button invokes standard Data link properties dialog. Advanced button is available for all connection types.

2. Data Sources

Every Data Source is connected to particular database connection. As was mentioned in previous chapters these objects are designed to periodically read database tables and display them in an OPC DA client. It supports two ways to read data from database:

- Direct access to database table (view),
- SQL command based on SELECT statement.



Underneath data sources the Data Items are defined (see chapter 3).

2.1 Direct Table Access

To configure Data Source object with direct table access the user have to fill in fully qualified database table name (or view). To locate specific table row(s) table parameters can be used. Parameters define subset of table columns used together as a filter for returned table row(s). Parameter values are specified by Data Items.

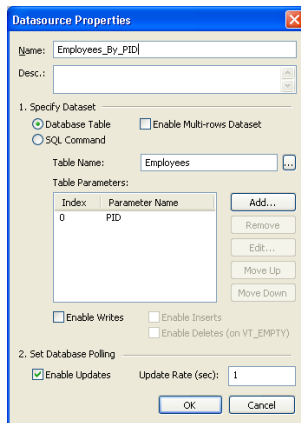


Figure 4

Enable Multi-rows Dataset

If user wants to read more than one row from database table (parameters doesn't identify rows uniquely) then "**Enable Multi-rows Dataset**" check box has to be switched on.

Enable Writes

Direct database access allows users to make modifications in specified database table (may work for views with limitations given by connected database server or engine). This feature is switched on by "**Enable Writes**" check box.

Note: Writes are disabled for multi-rows datasets.

Enable Inserts

When parameter values don't match any row in the table and inserts are enabled the new row is inserted into database table.

Enable Deletes

When VT_EMPTY value is written into a record set oriented data item and deletes are enabled then related row will be deleted from database table.

2.2 SQL Command

The SQL command data source is designed to access database data using SQL commands based on SELECT statements. The SQL command can be parameterized by SQL command parameters. Parameters in SQL command are represented by question marks (?) and they are filled in the same order as configured in "**SQL Command Parameters**" section.

Database stored procedures returning cursor variables (row sets) can be also accessed using SQL command based on EXECUTE or CALL statements.

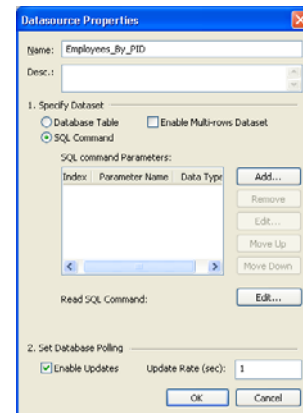


Figure 5

2.3 Database Polling

Database polling properties should be specified for any data source object. Specifying the Update Rate the user can set a minimum time period for scanning data sources by the Data Mining Server. When Updates are disabled then Data Mining Server scans the database only once when the first data item is requested or on the users request invoked by refresh data item.

3. Data Items for Data Source

Two types of Data Items are used under Data Sources:

- Statically defined (preconfigured)
- Dynamically requested (parameter values are set in runtime)

Note: Data Items are accessible as OPC DA tags; i.e. they are visible in OPC DA browsers as ordinary OPC DA Tags. The



ProgID of the Data Mining server is “*ICONICS.DatabaseOPCServer.3*”.

3.1 Static Data Items

Static Data Items have to be preconfigured and can't be changed in runtime. Basically there are used two types of Static Data Items:

Column oriented Data Item; gets value(s) from the table column specified in the configuration

Record set oriented Data Item; gets the whole table row(s) as an array of VARIANT values (single or multi dimensional array).

3.2 Dynamic Data Items

Dynamic Data Items are available under every data source object without being configured. The syntax of fully qualified OPC DA tag ID is as follows:

ICONICS.DatabaseOPCServer.3\DataSourceName<ParameterValues>.DataItemName

Where

DataSourceName is the name of the data source object in server configuration,

ParameterValues are values for data source parameters delimited with commas (see Table 1: Parameter Types),

DataItemName can be one of “<Column Name>”, “@@Recordset”, “@@Refresh”, “@@LastScanTime” (see Table 2: Data Item Capabilities for details).

Table 1: Parameter Types

Data Type	Description	Value Syntax
String (VT_BSTR)	String	“text”
Integer (VT_I4)	4-bytes integer	123
Real (VT_R8)	8-bytes floating-point number (double)	3.14 or -1.82E-3
Boolean (VT_BOOL)	Boolean	BOOL (True) or BOOL (False)
VT_DATE	8-byte floating-point number	DATETIME(date value)

Note: The DATE type is implemented using an double-precision numbers. Days are represented by whole number increments starting with 30 December 1899, midnight as time zero. Hour values are expressed as the absolute value of the fractional part of the number.

Table 2: Data Item Capabilities

Data Item Name	Access Rights	Description
@@Column	R/W	Can address the only one table column identified by its name placed into the last section of OPC DA tag name; delimited by <>.
@@Recordset	R/W	Addresses the whole table row.
@@Refresh	W	Any write forces the data source to re-read data from connected data source.
@@LastScanTime	R	Returns time of the last data source update.

Dynamic data items can be indexed by parameters configured at the data source level. It allows changing parameter values dynamically in the runtime, e.g. using aliases, variables or scripting.

3.3 Examples:

Let suppose the database table “Sample” with columns “PID”, “Name”, “Description” and data source with name “Sample_by_PID” referring to this table with one parameter “PID”. Table doesn't contain any row with PID higher then 5.

The OPC DA tag

“*ICONICS.DatabaseOPCServer.3\Sample_by_PID<1>.<Name>*” accesses the “Name” column for row with PID = 1.

The OPC DA tag

“*ICONICS.DatabaseOPCServer.3\Sample_by_PID<1>.@@Recordset*” accesses the whole table row with PID = 1. Table row(s) values are returned in array of VARIANT values (one or two dimensional – depends on data source configuration).

Inserts into column oriented data item configured to access table column “Name” with PID set 10:



Writing string “John” into this data item will create a new table row with the following values: PID = 10, Name = “John”, Description = database default value.
Inserts into record set oriented OPC tag with PID set 11:
Writing array of VARIANTS will create a new table row using all values from passed array (overrides database default values and parameter values)
Delete of table row with PID = 11:
Writing VT_EMPTY value into data item used in previous example will delete row with PID = 11.

4. Data Manipulators

Every data manipulator is connected to particular database connection. Data manipulator objects were designed to manipulate database data and structure using stored procedures or simple SQL commands when database server or engine does not support Stored Procedures. In comparison with data sources it should not return any cursor variables (row sets). Data manipulation actions are executed on the user request only (there is no polling mechanism).

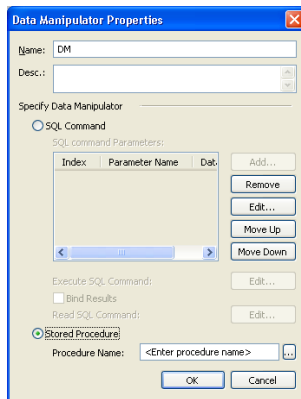


Figure 6

4.1 Data Items for Data Manipulator

There two types of Data Items that can access Data Manipulator objects:

- Execute data item,
- Result data item.

The data manipulator object executes its action asynchronously; unique request identification is needed to pair requests and responses.

To access data manipulator objects in synchronous way ICONICS ships with GENESIS-32 two components – DBOPCManipulatorWrapper and DBOPCManipulatorActiveX.

4.2 Stored Procedures

Stored Procedures are preferred way of using data manipulators. They are much more flexible then SQL commands and usually faster in processing. Their wide range of capabilities allows making complex database data manipulation.

Table 3: Data Item Capabilities

Data Item Name	Access Rights	Description
@@Execute	W	Writing to this data item executes data manipulator action (stored procedure or SQL command). The actual value written into this tag has to conform to the following rules: It has to be an array of VARIANTS The very first value in this array is the unique request identification The rest of the values are used as input parameters for data manipulator.
@@Result	R	Returned values represents responses for execute requests. All the returned values are stored in an array of VARIANTS. Similarly to @@Execute data item the very first value in this array is the unique request identification used in execute data item. The rest of the values are output parameters (if there are any).

4.3 SQL Command

The SQL command data manipulators are mainly designed for databases that don't support stored procedures (MS Excel, partly MS Access, etc.). It allows configuring two SQL commands – one for actual data manipulation action, the second for getting results (if needed). Setting the parameters is similar to SQL commands in the data source configuration. Result SQL command is using the same set of parameters as Execute SQL command in data manipulation action.



5. Frequently Asked Questions about Data Mining OPC server

When I select SQL Command under Data Source what it should do?

SQL command will behave similarly to table based data source - it will send data from record set returned by this command. You can parameterize it using data item parameters.

When I select enable multi select rows what it should do (what should I expect as the result)

You will get multiple rows from SQL query or table. In the case of single column it will return single-dimensional array of values where each value will represent one row. When you will use @@Recordset then you will get two-dimensional array (like table content) - all columns and rows.

Note: One limitation is "Multi-rows datasets" - is read only.

What are SQL Command parameters, why should I give more than one of these?

SQL command parameters specifying a set of parameters used by SQL command; SQL command can use more than one parameter.

Why and when should I give more than one table parameters?

You may want to locate a table row using different criteria - E.g. if you have "employees" table you can index it using "name, mid-name, surname" or you can use "personal ID".

When should I check Enable writes? How are the writes possible (like from GraphWorX)?

If you will use table based data source with single-row as output then every column oriented data item will return single value. In this case you can request it as process point and modify it.

After I check Enable Writes, when should I check Enable Inserts?

In the case the related table row is missing in the table then after writing it will create this table row using combination of known parameters and database default values.

What a Data Manipulator is and what it does?

Data manipulator object allows you to execute stored procedures or standalone SQL queries and get results from them (if there are any). There are two special data items to manipulate it

@@Execute and
@@Results.

You can execute stored procedure by writing into @@Execute data item where the value should be safe array of VARIANT values. Very first value is "request identification" the rest of the values are input parameters for this procedure. You will get results from @@Result data item. The value have very similar structure as the execute one. It is again an array of VARIANT values where the very first value is the "request identification" and the rest of the values are output parameters.

What does Bind Results for SQL Command do, why is the Edit button available. What should I be able to do with this button?

"SQL Command" section in data manipulator object was design for databases (like MS Access), which doesn't have stored procedures support implemented (or have it very poor). Then the only way how to manipulate the data is to use SQL commands. Then one command is used to do specific action and a result SQL command is used to get output values in the same database transaction. E.g. when you insert a new table row into MS Access database you can then read an auto-generated value from the table using "SELECT @@identity" command. The other sample can be if you will insert/update something then you can get back a sum of certain column(s). The range of its usage is very wide and depends on specific implementation.

Where should be stored procedures written? (For example if I connect to SQL Server, should the stored procedure be written in SQL and already available for us to use it)

Stored procedures are stored in database server and allow you to manipulate database content. They are usually faster then using SQL commands from client side. To get more info you will need to study specific database server doc.

What should I expect when I check, complete record set.

When you check "complete records set (returned as array)" check box it will return a single row data (all columns) in single-dimensional array. When "multi-rows dataset" will be enabled you will get multi-dimensional array as result (similarly to "@@Recordset" data item).

6. Do's and Don'ts for Data Mining

Data Mining OPC Server

Not to Do:

- Do not use Store Procedures in Data Manipulators with MS Access databases.
- Do not use spaces in stored procedures names.



- Do not use CREATE, INSERT, UPDATE statements in Data Source SQL Command.
- Do not use Data Manipulator stored procedures to view data.

To Do:

- Use stored procedures in Data Manipulators with MS SQL and Oracle.
- Use SQL Queries when connecting to MS Excel or MS Jet Connections.
- Use Data Manipulator stored procedures to manipulate data only.

Data Mining Grid

- Cannot see single dimensional array (which is a row) as a Row in DB Grid, can be only seen as a Column.
- Do not try to use Data Mining tags in ADO.